Solaris Management Facility (SMF) - Workshop

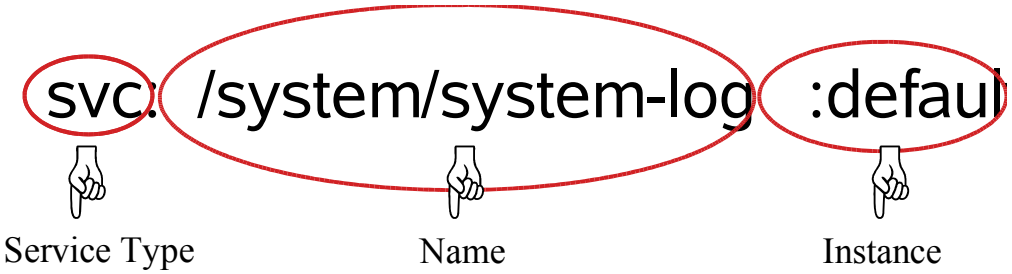
Ganesh Hiregoudar
Renaud Manus
OP/N1 RPE Approachability
Sun Microsystems

Thanks!

- We would like to thanks the following engineers who participated in writing and delivering this SMF workshop.
  - Jarod Nash,
  - Jason Banham,
  - Lee Brooks,
  - Rhodri Davies,
  - Phill Hughes

Terminology

- Service
  - > Object (application, software state of a device, set of other services) that can be managed and observed.
- Instance
  - > Child of service object.
- FMRI (Fault Managed Resource Identifier)
  - > 3 components:



- Restarter
  - > Service responsible for restarting service(s)

Terminology (cont.)

- Dependency
  - Formal description of the other services that are required to start a service
- Milestone
  - Predefined set of capabilities for a set of services.
- Contract
  - Mechanism within the kernel for restarters to stay informed about the service instances they start.

Terminology (cont.)

- Manifest
  - Description and initial configuration file for a service. Delivered with the product in the form of an XML file.
- Repository
  - Configuration database for all services. Allows setting to remain persistent across reboots.
- Snapshot
  - Historical data about each service instance in the repository for administrative inspection and rollback.

Key files and directories

- /var/svc/manifest/*
- /lib/svc/method/*
- /var/svc/log/*
- /var/svc/profile/*
- /etc/svc/repository.db
  - Global configuration database
- /lib/svc/see/global.db
  - Minimal configuration database

New filesystems

- /etc/svc/volatile
  - Contains transient data, eg. lock files, init state and some log files. Allocated from swap.
- /system/contracts
  - Primary interface into contract subsystem, to allow service instances to be restarted. Indirectly related to SMF.

SMF commands

- svcs : report the status of service instances
- svcadm : manipulate services instances
  - > eg. enable, disable, restart, ...
- svccfg : manipulate the repository
  - > eg. import, export, configure, ... services
- svcprop : display properties for a given service instance
- inetadm : report and configure inetd based services
- inetconv : convert and import services from inetd.conf

Commands: svcs(1)

- List enabled or all (-a) instances, sorted by state, time
- Explanations for errors/states (-x)
- Show dependencies (-d) and dependents (-D)
- Show member processes (-p), additional details (-v/-l)

```
$ svcs
STATE          STIME    FMRI
....
online         18:18:30 svc:/network/http:apache2
online         18:18:29 svc:/network/smtp:sendmail
....
$ svcs -p sendmail
STATE          STIME    FMRI
online         18:18:29 svc:/network/smtp:sendmail
               18:18:29   100180 sendmail
               18:18:29   100181 sendmail

$ svcs -d sendmail
STATE          STIME    FMRI
online         18:17:44 svc:/system/identity:domain
online         18:17:52 svc:/network/service:default
....
```

Commands: svcs(1)

- You can use a pattern which uses globbing rules to build regular expressions
- svc:/ is automatically added to the pattern if omitted
- Examples:

```
$ svcs '*print*'
$ svcs cron
$ svcs dns/client
```

Commands: svcadm(1M)

- Enable, disable, refresh, restart service instances
- Mark in special states (maintenance)
- Synchronously wait for changes (-s)

```
$ grep lianep /etc/user_attr
lianep::::auths=solaris.smf.modify,solaris.smf.manage
$ svcs apache2
STATE          STIME    FMRI
-              ?        svc:/network/http:apache2
$ # create /etc/apache2/httpd.conf
$ svcadm enable apache2
STATE          STIME    FMRI
online         19:19:01 svc:/network/http:apache2
$ # edit /etc/apache2/httpd.conf
$ svcadm refresh apache2
$ svcs apache2
STATE          STIME    FMRI
online         19:19:33 svc:/network/http:apache2
$ svcadm disable apache2
$ svcs apache2
STATE          STIME    FMRI
disabled       19:20:07 svc:/network/http:apache2
```

See also http://blogs.sun.com/roller/page/sch/20040923

Commands: svcadm(1M) actions

- **enable**: allow start once dependencies are satisfied
- **disable**: stop service and do not allow it to start again
  - -t: enable/disable until the system is rebooted
  - -s: enable/disable synchronously (wait for it...)
- **refresh**: reload service configuration and run the refresh method (if any)
- **restart**: stop the service, then allow it to start once its dependencies are satisfied (no configuration change made)
- **clear**: mark service as repaired

Commands: `svcadm(1M)` milestone

- Milestone: A service which specifies a collection of dependencies which declare a specific state of system-readiness
  - `init S, boot -s` and friends still work
- Major milestones, which are analogous to system run-levels can be reached directly
  - from boot (`-m milestone=`),
  - the standard `init` invocations,
  - or via `svcadm: milestone/single-user, milestone/multi-user, milestone/multi-user-server`

See also http://blogs.sun.com/roller/page/lianep/20050204

Commands: svcprop(1)

- List properties of services and instances
- Fetch in convenient forms for scripting
- View running or current props (-c), uncomposed (-C)

```
$ svcprop network/http:apache2
...
physical/entities fmri svc:/network/physical:default
physical/grouping astring optional_all
physical/restart_on astring error
physical/type astring service
start/exec astring /lib/svc/method/http-apache2\ start
start/timeout_seconds count 60
start/type astring method
stop/exec astring /lib/svc/method/http-apache2\ stop
stop/timeout_seconds count 60
stop/type astring method
restarter/auxiliary_state astring none
restarter/next_state astring none
restarter/state astring disabled
restarter/state_timestamp time 1102030556.737590000

$ svcprop -p enabled network/http:apache2
false
```

Administration exercise

- Write a script which:
  - prints the start/exec property for sendmail
  - enables sendmail
  - waits for the service to start
  - prints the service state

Administration exercise

- Write a script which:
  - prints the start/exec property for sendmail
  - enables sendmail
  - waits for the service to start
  - prints the service state

```
#!/bin/ksh

/usr/bin/svcprop -p start/exec svc:/network/smtp:sendmail

/usr/sbin/svcadm enable -s svc:/network/smtp:sendmail
if [ $? -ne 0 ]; then
     echo sendmail failed to start
fi

/usr/bin/svcprop -p restarter/state svc:/network/smtp:sendmail
```

Commands: inetadm(1M)

- List services managed by inetd
- View (-l) and modify (-m) inetd-specific properties

```
$ inetadm
...
enabled   online         svc:/network/ftp:default
enabled   online         svc:/network/finger:default
disabled  disabled       svc:/network/login:eklogin
disabled  disabled       svc:/network/login:klogin
enabled   online         svc:/network/login:rlogin
disabled  disabled       svc:/network/rexec:default
enabled   online         svc:/network/shell:default

$ inetadm -l ftp
SCOPE    NAME=VALUE
         name="ftp"
         endpoint_type="stream"
         proto="tcp6"
         isrpc=FALSE
         wait=FALSE
         exec="/usr/sbin/in.ftpd -a"
         user="root"
...
default  tcp_wrappers=FALSE
```

svccfg(1M)

svccfg manipulates the repository

Uses sub-commands to perform actions

Common sub-commands

select <fmri>	: select a service / instance
list	: show children of the selected service
listprop	: display the properties for a service
setprop	: change a property value for a service
delete	: delete a service / instance
validate <file>	: validate an XML manifest file
import <file>	: import a manifest file into repository
unselect	: navigate to parent of current selection

Commands: svccfg(1M)

- Import, export manifests; apply, extract profiles
- Interactive mode for modifying properties

```
$ svccfg -v import /var/svc/manifest/network/http-apache2.xml
svccfg: Refreshed network/http:/apache2
svccfg: Successful import.


$ svccfg
svc:> select network/http:apache2
svc:/network/http:apache2> listprop
...
general                    framework
general/enabled            boolean  false
...
start                      method
start/exec                 astring  "/lib/svc/method/http-apache2 start"
start/timeout_seconds      count    60
start/type                 astring  method
svc:/network/http:apache> editprop
[$EDITOR launches, allows direct editing of properties]
svc:/network/http:apache2> exit

$ svcadm refresh apache2      # read changed config
$ svcadm restart apache2      # restart with changed config
```

svccfg(1M): example (cont.)

- To change a property value:
  - Use the listprop sub-command to view properties
  - Use the setprop sub-command to set values

Example:

Increase the file descriptor limit for lpsched

```
# svccfg
   svc:> select application/print/server
   svc:/...> listprop
   svc:/...> setprop lpsched/fd_limit = 8192
   svc:/...> quit
# svcadm refresh print/server:default
# svcadm restart print/server:default
```

svcs - example

Checking dependencies (NFS server)

```
# svcs -d svc:/network/nfs/server:default
STATE          STIME    FMRI
online         Nov_15   svc:/network/loopback:default
online         Nov_15   svc:/network/physical:default
online         Nov_15   svc:/network/rpc/bind:default
online         Nov_15   svc:/network/rpc/keyserv:default
online         Nov_15   svc:/system/filesystem/local:default
online         Nov_15   svc:/network/rpc/gss:ticotsord
online         Nov_15   svc:/network/nfs/mapid:default
online         10:32:25 svc:/network/nfs/nlockmgr:default
```

inetadm - examples

Changing properties of an inetd service (in.ftpd)

```
# inetadm -l svc:/network/ftp:default
SCOPE    NAME=VALUE
         name="ftp"
         endpoint_type="stream"
         proto="tcp6"
         isrpc=FALSE
         wait=FALSE
         exec="/usr/sbin/in.ftpd -a"
         user="root"
            .
            .
            .
# inetadm -m svc:/network/ftp:default exec="/usr/sbin/in.ftpd -a -l"

# inetadm -l svc:/network/ftp:default | grep exec
         exec="/usr/sbin/in.ftpd -a -l"
```

Misc - examples

Milestone transition: None to Single User

```
ok boot -m milestone=none
...
# svcs '*milestone*'
STATE          STIME    FMRI
disabled       13:12:49 svc:/milestone/multi-user:default
disabled       13:12:49 svc:/milestone/multi-user-server:default
disabled       13:13:04 svc:/milestone/single-user:default
[edited for brevity]

# svcadm milestone svc:/milestone/single-user:default

# svcs | grep '\*'
offline*       13:32:26 svc:/milestone/single-user:default
offline*       13:32:28 svc:/system/device/local:default
```

Advanced concepts

- Service instance states
- Repository
- Manifest
- Snapshots
- Profiles
- Demo

Service instance states

Uninitialised The service instance is not running and the configuration data has not been read. Usually seen if you have booted to milestone=none

Offline The service instance is not running although the configuration data has been read. This is usually the result of a dependency that has not been met, or if there is an error in the start method.

Online The service instance is running. All of its dependencies have been met.

Disabled A service instance is not running. This may be the default state for a service when it is first imported into the repository, or an administrator may have marked the service as disabled. It will require operator assistance to move out of this state.

Degraded A limited set of failures (usually dependencies) may cause the service instance to function in a limited capacity, eg: if an inetd service can work with IPv4 and IPv6 but the latter is not configured, then only IPv4 will be in use and thus the service may be in a degraded state.

Maintenance The service instance is unavailable due to an error. There are many reasons why, ranging from unsatisfied dependencies and failed start methods, to more complicated reasons.

Repository

- The repository is THE source for all known services on the system
- Modifications to service configuration and state are stored in the repository
- Services live in the repository!
- The repository is an SQLite database

Manifest

- DTD is the reference to write manifest
  - /usr/share/lib/xml/dtd/service_bundle.dtd.1
- Explore /var/svc/manifest for similar services
  - system/utmp is a simple standalone deamon
  - system/coreadm is a simple configuration service
  - network/telnet is an inet-managed deamon
- Initial inet service manifests can be created easily by invoking “inetconv -i <file>”

Manifest (order is important)

- Service bundle
- Service name
- Default instance
- Single instance
- Dependencies
- Exec. Methods
- Property groups
- Stability level
- Template

Manifest (cont.)

- Basic elements
  - Name of the service
  - Number of instances
  - Start, stop, [refresh] methods
  - Dependencies
- Advanced elements
  - Property groups
  - Service model
  - Fault handling section
  - Documentation template

Service relationships

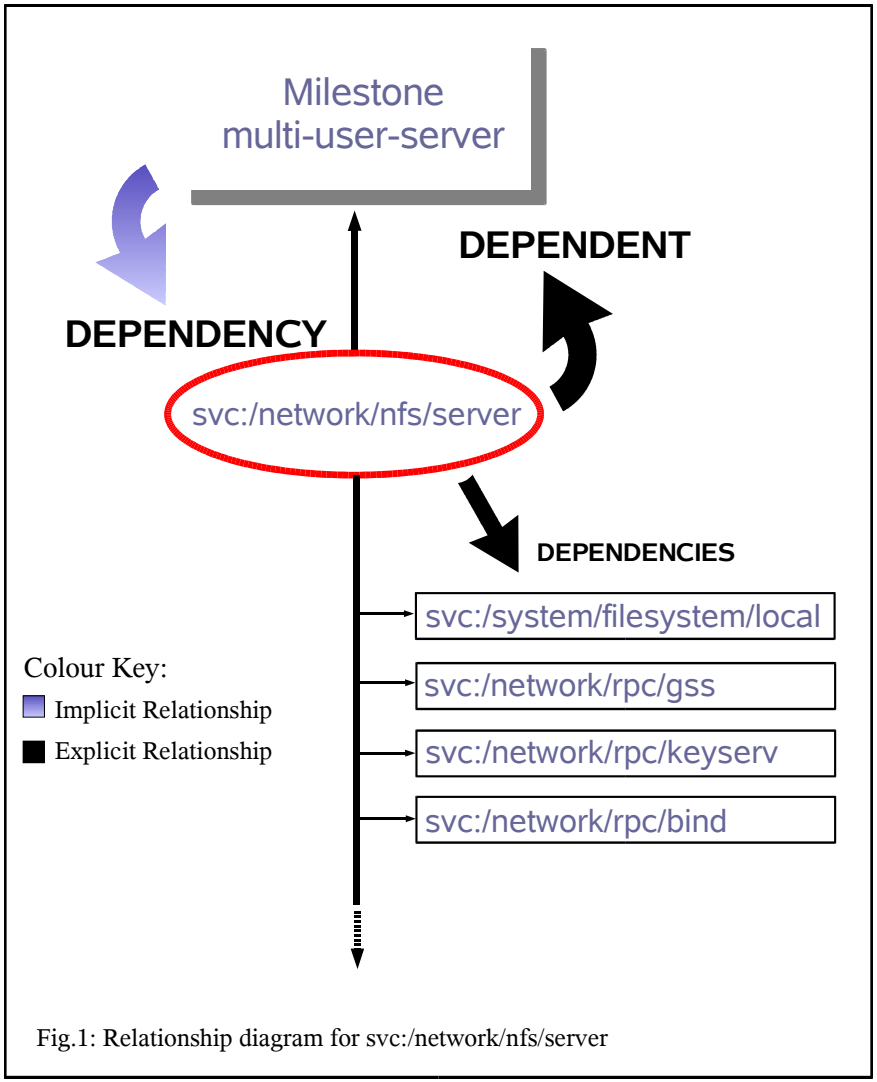


Fig.1: Relationship diagram for svc:/network/nfs/server

Dependency

The other services that a service relies upon in order to start

Dependent

A mechanism for a service to declare itself as a dependency of another service

Service relationships

- Restart relationships defined by dependency groups
  - grouping
    - require_all : all services are running
    - require_any: at least one service is running
    - optional_all : all services running, disabled or maintenance
    - exclude_all : all services disabled, maint. or absent
  - restart_on
    - none : required only for startup
    - error : restart if dependency fails due to hw or sw error
    - restart : restart if dependency restart to any reason
    - refresh : restart if dependency restarts or is refreshed

Manifest: dependents

- A service manifest can specify which services are dependent upon it.
- It is forbidden to modify the manifest of other services
- For predictable startup behaviour, the service manifest should specify one of the major milestones as a dependent.

```
<dependent
    name='nfs-client_multi-user'
    grouping='optional_all'
    restart_on='none'>
    <service_fmri value='svc:/milestone/multi-user' />
</dependent>
```

Development: utmpd(1M) example

```
<service name='system/utmp' type='service' version='1'>
    <create_default_instance enabled='true' />
    <single_instance />”
        <dependency name='milestone' grouping='require_all'
            restart_on='none' type='service'>
            <service_fmri value='svc:/milestone/sysconfig'/>
        </dependency>
        <dependent name='utmpd_multi-user' grouping='optional_all'
            restart_on='none'>
            <service_fmri value='svc:/milestone/multi-user'/>
        </dependent>

        <exec_method type='method' name='start'
            exec='/lib/svc/method/svc-utmpd' timeout='60' />
        <exec_method type='method' name='stop'
            exec=':kill' timeout='60' />
    <stability value='Unstable' />
    <template>
        <common_name><loctext xml:lang='C'>
            utmpx monitoring
        </loctext></common_name>

        <documentation>
        <manpage title='utmpd' section='1M'
            manpath='/usr/share/man' />
        </documentation>
    </template>
</service>
```

Snapshots

- A snapshot is a copy of the configuration information for a service instance
- Can't be used for 'cloning'
- Up to 5 levels of snapshot
  - initial : initial conf. delivered by a package
  - last-import: conf. of an instance when last imported by svccfg(1M)
  - running : the running conf. of the instance
  - start : conf. captured during a successful transition to online state
  - previous : conf. captured when an undo operation is performed

Snapshots (cont.)

- Snapshots taken at the instance level
- Most configuration settings at service level
- Use svccfg(1M) to select an instance
  - listsnap : list available snapshots
  - selectsnap : select one of the snapshots
  - revert [snap] : revert back to a given snapshot
- The changed property values can be made active via 'svcadm refresh'

Snapshots: example

```
[...]
svc:/network/smtp:sendmail> listsnap
initial
running
start
svc:/network/smtp:sendmail> selectsnap running
[running]svc:/network/smtp:sendmail> listprop config-file/*
config-file/entities    fmri     file://localhost/etc/mial/sendmail.cf
config-file/grouping    astring  require_all
config-file/restart_on  astring  refresh
config-file/type        astring  path
[running]svc:/network/smtp:sendmail> selectsnap start
[start]svc:/network/smtp:sendmail> listprop config-file/*
config-file/entities    fmri     file://localhost/etc/mail/sendmail.cf
config-file/grouping    astring  require_all
config-file/restart_on  astring  refresh
config-file/type        astring  path
[start]svc:/network/smtp:sendmail> revert
svc:/network/smtp:sendmail> listsnap
initial
running
start
previous

# svcadm refresh svc:/network/smtp:sendmail
```

Profiles

- A profile is a description of the services that are to be used on a system
  - Applied in order : generic, platform, site
  - Profiles may include sub-profiles
- Enable/Disable services for your site
  - eg. jumpstart, JASS JumpStart Architecture and Security Scripts
- Profile location : /var/svc/profile/*
- Never modify existing profiles
- The site.xml file is for local customizations, which you must create

Profiles (cont.)

- To create your own site.xml file:

```
svccfg:> extract > /var/tmp/site.xml
(modify file with your changes)
# cp /var/tmp/site.xml /var/svc/profile
# init 6
```

- You can also “svccfg apply” the site.xml for immediate effect
- Be aware, profiles are only processed once

Debug & Lab

- Troubleshooting techniques
- Lab : Writing your own service
- Lab exercises

Troubleshooting

- Service failures printed to console, syslog
- Always start with `svcs -x`; http://sun.com/msg
  - `svcs -x` will display the service logfile, if it exists:
    - `/var/svc/log`
    - `/etc/svc/volatile`
- See service start messages with `boot -m verbose`
- For a system that hangs during boot:
  - `boot -m verbose,milestone=none`
  - log in at prompt
  - `svcadm milestone all`
  - Watch system progress with `svcs`

Troubleshooting (cont.)

- To truss a service starting up
  - > Truss the restarter (svc.startd/inetd)
- Or
  - > Make a note of the start method
  - > Use svccfg to modify the start method
  - > When complete, restore the original start method

```
# svccfg
svc:> select application/print/rfc1179
svc:...1179> listprop inetd_start/exec
inetd_start/exec  astring  /usr/lib/print/in.lpd
svc:...1179> setprop inetd_start/exec =
"/usr/bin/truss -ealfo /tmp/in.lpd.truss /usr/lib/print/in.lpd"
svc:...1179> end
# svcadm refresh svc:/application/print/rfc1179:default
```

Troubleshooting (cont.)

- Maintenance mode
  - svcs -xv or svcs -l to see why the service failed
  - svcadm refresh <fmri> if repository has been changed
  - svcadm clear <fmri> once fault rectified
- Have changes been made to repository?
  - How many services are affected?
  - What snapshots do exist?
  - Can you revert back to a previous snapshot?

Troubleshooting (cont.)

- If the current repository is beyond repair, try to restore from an earlier repository or reseed:

```
/lib/svc/bin/restore_repository
```

- Will the system boot in verbose mode?

```
> ok boot -m verbose
```

- Will the system boot with no service?

```
> ok boot -m milestone=none
```

Lab – Unable to resolve webcache

Run the test1.breakme script

```
# nslookup webcache
 ;; connection timed out; no servers could be reached
 # svcs svc:/network/dns/client:default
 offline        17:34:25 svc:/network/dns/client:default
```

Lab – Unable to resolve webcache

```
# svcs -xv svc:/network/dns/client:default
svc:/network/dns/client:default (DNS resolver)
 State: offline since Fri Feb 04 18:10:24 2005
Reason: Dependency file://localhost/etc/resolv.conf is absent.
   See: http://sun.com/msg/SMF-8000-E2
   See: man -M /usr/share/man -s 3RESOLV resolver
   See: /var/svc/log/network-dns-client:default.log
Impact: This service is not running.
```

In this case it clearly tells us that /etc/resolv.conf is missing

```
# svcadm refresh svc:/network/dns/client:default
```

Lab – sendmail won't start

Run the test2.breakme script

```
svc.startd[7]: network/smtp:sendmail failed
# svcs svc:/network/smtp:sendmail
STATE          STIME    FMRI
maintenance    16:18:03 svc:/network/smtp:sendmail
```

Lab – sendmail won't start

```
# svcs -xv svc:/network/smtp:sendmail
svc:/network/smtp:sendmail (sendmail SMTP mail transfer agent)
 State: maintenance since Mon Nov 22 16:18:03 2004
Reason: Start method failed repeatedly, last died on Killed (9).
   See: http://sun.com/msg/SMF-8000-KS
   See: man -M /usr/man -s 1M sendmail
   See: /var/svc/log/network-smtp:sendmail.log
Impact: This service is not running.


# tail /var/svc/log/network-smtp:sendmail.log
[ Nov 22 16:17:53 executing start method ("/lib/svc/method/smtp-sendmail start") ]
[ Nov 22 16:18:03 Method or service exit timed out.  Killing contract 122 ]
[ Nov 22 16:18:03 Method "start" failed due to signal Killed ]
```

Lab – sendmail won't start

Knowing the start method timed out we can examine the repository for this service:

```
# svcprop -p start svc:/network/smtp:sendmail
start/exec astring /lib/svc/method/smtp-sendmail\ start
start/type astring method
start/timeout_seconds count 5
```

Having extracted this value we can compare it with the value in the manifest:

```
# cd /var/svc/manifest/network
# more smtp-sendmail.xml
<?xml version="1.0"?>
...
          <exec_method
                type='method'
                name='start'
                exec='/lib/svc/method/smtp-sendmail start'
                timeout_seconds='120' />
```

Lab – sendmail won't start

1) Use svccfg to set the correct start method timeout:

```
# svccfg
svc:> select svc:/network/smtp:sendmail
svc:/network/smtp:sendmail> setprop start/timeout_seconds = 120
svc:/network/smtp:sendmail> quit
# svcadm refresh svc:/network/smtp:sendmail
# svcadm clear svc:/network/smtp:sendmail
# svcs svc:/network/smtp:sendmail
STATE          STIME    FMRI
online         17:03:48 svc:/network/smtp:sendmail
```

Lab – sendmail won't start

2) Use snapshots to set the correct start method timeout:

```
# svccfg
svc:> select svc:/network/smtp:sendmail
svc:/network/smtp:sendmail> selectsnap initial
[initial]svc:/network/smtp:sendmail> revert
svc:/network/smtp:sendmail> quit
# svcadm refresh svc:/network/smtp:sendmail
# svcadm clear svc:/network/smtp:sendmail
# svcs svc:/network/smtp:sendmail
STATE          STIME    FMRI
online         17:03:48 svc:/network/smtp:sendmail
```

Lab – keyserv won't start

Run the test3.breakme script

```
svc.startd[7]: svc:/network/nis/client:default:
Method "/lib/svc/method/yp" failed with exit status 96.

svc.startd[7]: svc:/network/rpc/keyserv:default:
Method "/usr/sbin/keyserv" failed with exit status 96.
# svcs svc:/network/nis/client:default
STATE          STIME    FMRI
maintenance    10:52:40 svc:/network/nis/client:default
# svcs svc:/network/rpc/keyserv:default
STATE          STIME    FMRI
maintenance    10:52:40 svc:/network/rpc/keyserv:default
```

Lab – keyserv won't start

```
# svcs -xv network/rpc/keyserv:default
svc:/network/rpc/keyserv:default (RPC encryption key storage)
 State: maintenance since Wed Feb 09 17:20:46 2005
Reason: Start method exited with SMF_EXIT_ERR_CONFIG.
   See: http://sun.com/msg/SMF-8000-KS
   See: keyserv(1M)
   See: /var/svc/log/network-rpc-keyserv:default.log
Impact: This service is not running.

# svcs -xv network/nis/client:default
svc:/network/nis/client:default (NIS (YP) client)
 State: maintenance since Wed Feb 09 17:20:46 2005
Reason: Start method exited with $SMF_EXIT_ERR_CONFIG.
   See: http://sun.com/msg/SMF-8000-KS
   See: ypstart(1M)
   See: /var/svc/log/network-nis-client:default.log
Impact: This service is not running.
```

Lab – keyserv won't start

```
# cd /var/svc/log
# tail network-rpc-keyserv:default.log
[ Feb  9 17:17:19 Stopping because service disabled. ]
[ Feb  9 17:17:20 Executing stop method (:kill) ]
[ Feb  9 17:20:45 Executing start method ("/usr/sbin/keyserv") ]
[ Feb  9 17:20:46 Method "start" exited with status 96 ]

# tail network-nis-client:default.log
[ Feb  9 17:17:19 Stopping because service disabled. ]
[ Feb  9 17:17:20 Executing stop method (:kill) ]
[ Feb  9 17:20:45 Executing start method ("/lib/svc/method/yp") ]
/lib/svc/method/yp: domainname not set
[ Feb  9 17:20:46 Method "start" exited with status 96 ]



# ls -l /etc/defaultdomain
/etc/defaultdomain: No such file or directory
```

Lab – keyserv won't start

```
# cat > /etc/defaultdomain
louvre.France.Sun.COM
^D
# domainname `cat /etc/defaultdomain`

# svcadm clear nis/client:default
# svcs nis/client:default
STATE        STIME   FMRI
online        9:59:21 svc:/network/nis/client:default

# svcadm clear network/rpc/keyserv:default
# svcs network/rpc/keyserv:default
STATE      STIME    FMRI
online     10:01:46 svc:/network/rpc/keyserv:default
```

Lab – customer ndd script

Run the test4.breakme script

```
Feb 10 12:25:21 labhost svc.startd[7]: network/ndd-nettune:default failed repeatedly


# svcs svc:/network/ndd-nettune:default
STATE          STIME    FMRI
maintenance    16:02:26 svc:/network/ndd-nettune:default
```

Lab – customer ndd script

```
# svcs -xv network/ndd-nettune:default
svc:/network/ndd-nettune:default (ndd network tuning)
 State: maintenance since Thu Feb 10 12:25:21 2005
Reason: Restarting too quickly.
   See: http://sun.com/msg/SMF-8000-L5
   See: man -M /usr/share/man -s 1M ndd
   See: /var/svc/log/network-ndd-nettune:default.log
Impact: This service is not running.

# tail /var/svc/log/network-ndd-nettune:default.log
[ Feb 11 15:12:14 Executing start method ("/lib/svc/method/ndd-nettune") ]
/sbin/sh: /lib/svc/method/ndd-nettune: cannot execute
[ Feb 11 15:12:14 Stopping because all processes in service exited. ]
[ Feb 11 15:12:14 Executing start method ("/lib/svc/method/ndd-nettune") ]
/sbin/sh: /lib/svc/method/ndd-nettune: cannot execute
[ Feb 11 15:12:14 Stopping because all processes in service exited. ]
[ Feb 11 15:12:14 Executing start method ("/lib/svc/method/ndd-nettune") ]
/sbin/sh: /lib/svc/method/ndd-nettune: cannot execute
[ Feb 11 15:12:14 Stopping because all processes in service exited. ]
[ Feb 11 15:12:14 Restarting too quickly, changing state to maintenance ]
```

Lab – customer ndd script

```
# ls -l /lib/svc/method/ndd-nettune
-rw-r--r--   1 root     root         477 Feb 11 15:12 /lib/svc/method/ndd-nettune

# chmod 755 /lib/svc/method/ndd-nettune
# svcadm clear network/ndd-nettune:default
Feb 11 15:19:51 labhost svc.startd[7]: network/ndd-nettune:default failed
repeatedly

# more /var/svc/log/network-ndd-nettune:default.log
[ Feb 11 15:19:48 Leaving maintenance because clear requested. ]
[ Feb 11 15:19:49 Enabled. ]
[ Feb 11 15:19:49 Executing start method ("/lib/svc/method/ndd-nettune") ]
[ Feb 11 15:19:49 Stopping because all processes in service exited. ]
[ Feb 11 15:19:49 Executing start method ("/lib/svc/method/ndd-nettune") ]
[ Feb 11 15:19:49 Stopping because all processes in service exited. ]
[ Feb 11 15:19:49 Executing start method ("/lib/svc/method/ndd-nettune") ]
[ Feb 11 15:19:49 Stopping because all processes in service exited. ]
[ Feb 11 15:19:51 Restarting too quickly, changing state to maintenance ]
```

Lab – customer ndd script

```
# cd /lib/svc/method
# file ndd-nettune
ndd-nettune:    executable /sbin/sh script

# more /lib/svc/method/ndd-nettune
#!/sbin/sh
#
# ident   "@(#)ndd-nettune.xml    1.0     04/09/21 SMI"

. /lib/svc/share/smf_include.sh
. /lib/svc/share/net_include.sh

# Make sure that the libraries essential to this stage of booting can be found.
LD_LIBRARY_PATH=/lib; export LD_LIBRARY_PATH
echo "ndd ran" >> /tmp/smf.out
/usr/sbin/ndd -set /dev/tcp tcp_recv_hiwat 16384
/usr/sbin/ndd -set /dev/tcp tcp_xmit_hiwat 16384
# Reset the library path now that we are past the critical stage
unset LD_LIBRARY_PATH
```

Lab – customer ndd script

A transient script should run once then exit and never be restarted.
In the log file you can see that the start method is being run repeatedly.
This suggests the service has been configured incorrectly, so you should look at the repository:

```
# svccfg
 svc:> select network/ndd-nettune
 svc:/network/ndd-nettune> listprop startd/*
 startd/duration  astring  child
```

Here you see this service has been mis-configured as a child service, so when the process died svc.startd attempted to restart it. We know it should be transient so:

```
 svc:/network/ndd-nettune> setprop startd/duration = transient
 svc:/network/ndd-nettune> quit
 # svcadm refresh svc:/network/ndd-nettune:default
 # svcadm clear svc:/network/ndd-nettune:default
 STATE          STIME    FMRI
 online         15:20:18 svc:/network/ndd-nettune:default
```

Service Lifecycle

- Determine service suitability
- Write service methods
- Write service manifest
- Import manifest
- Test/modify
- Package manifest and methods
- Install service
- Service startup/runtime
- Configure/modify service
- Remove service

Method creation

- Start with `/etc/init.d` script
- Include `/lib/svc/share/smf_include.sh`
  - `SMF_EXIT_OK`
  - `SMF_EXIT_ERR_FATAL`
  - `SMF_EXIT_ERR_CONFIG`
- Fail with a non-0 exit code, and emit a message to `stdout` or `stderr`
- `start` and `stop` required, `refresh` optional
- `:kill [signal]` and `:true`
- Don't exit until service is completely started
- remove script from `/etc/rc?.d` locations and `/etc/init.d`

Write your own service

- Tip : How to check the manifest?
  - > Use xmllint(1) or Mozilla to parse the manifest and check for errors
  - > Example:

```
# xmllint mysvc.xml
   mysvc.xml:71: parser error : Opening and ending tag mismatch:
                                exec_method line 36 and service </service>
                                                                ^
   mysvc.xml:73: parser error : expected '>' </service_bundle>
                                              ^
   mysvc.xml:74: parser error : Premature end of data in tag service_bundle
                                line 10
```

Write your own service

- `littled` : A simple daemon that listens on port 13567
  - Requires a configuration file: `/var/tmp/littled.conf`
- You can interact with **littled** thus:

```
$ telnet localhost 13567
SMF TOI littled daemon/server version=0.8
Daemon Started: Wed Feb 09 11:07:44 2005
littled> help
SMF TOI littled daemon/server version=0.8
Available commands:
      help - Show help information
       bye - Disconnect from daemon
       die - Kill daemon
    status - Damon Status Information
 prtconfig - Display configuration
    signal - Die horribly via a signal
readconfig - Re-read configuration file
```

Write your own service

- Your manifest must include:
  - A name
  - A default instance
  - A start and stop method
  - A list of dependencies
- Optional:
  - A template/documentation section

Write your own service

- Write a manifest for the service
  - You can use /var/svc/manifest/system/utmp.xml as a template
- Import it into the repository
  - Validate it first
  - If you need to make corrections you may need to delete the service before re-importing
  - Use listprop to look at your service in the repository
- Make sure you can start and stop it
- Where is the log file for the service?
- Check littled restarts by killing its PID

Write your own service

- Open a terminal and telnet to port 13567
  - Display the configuration for littled
  - Run `date >> /var/tmp/littled.conf`
  - Send the PID for littled a HUP signal
  - Redisplay the configuration
- Using this knowledge write a refresh method for littled
- Disable the **littled** service
  - Does your telnet connection exit?

References

- Jarod Nash's workshop material
- Additional quickstart and developer documentation available at
  http://www.sun.com/bigadmin/content/selfheal/
- Solaris System Administration Guide (chapters 9 and 14):
  http://docs.sun.com/app/docs/doc/817-1985
- Blogs:
  http://blogs.sun.com/sch
  http://blogs.sun.com/lianep
- Internally: http://greenline.sfbay
- Subscribe to greenline-interest@sun.com
- http://slp.sun.com/sun
  WZI-SS-3204 : Fault & Service Management WBT
- SMF Community page http://communities.uk/solaris/smf/

The End

ganesh.hiregoudar@sun.com
renaud.manus@sun.com
http://blogs.sun.com/ganesh

Additional slides for lab

Lab - Unable to print

Run the test5.breakme script

```
# lp -dlpdummy /etc/hosts
UX:lp: ERROR: Can't establish contact with the LP print service.
      TO FIX: Either the LP print service has stopped,
              or all message channels are busy. If the
              problem continues, get help from your
              system administrator.

# svcs -p svc:/application/print/server:default
STATE          STIME    FMRI
disabled       11:59:47 svc:/application/print/server:default
```

Lab - Unable to print

```
# svcs -xv svc:/application/print/server:default
svc:/application/print/server:default (LP print server)
 State: disabled since Fri Feb 11 16:10:37 2005
Reason: Temporarily disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-1S
   See: man -M /usr/share/man -s 1M lpsched
   See: /var/svc/log/application-print-server:default.log
Impact: 1 dependent service is not running:
        svc:/application/print/ipp-listener:default
```

Lab - Unable to print

```
# man -s 1M lpsched

System Administration Commands                          lpsched(1M)
NAME
     lpsched - start the LP print service
SYNOPSIS
     lpsched  [-f num_filters]  [-n num_notifiers]  [-p fd_limit]
     [-r reserved_fds]
DESCRIPTION
     The  lpsched command  starts or restarts the  LP print  ser-
     vice.

     The  lpshut command stops the  LP print  service.  Printers
     that are restarted using lpsched reprint (in their entirety)
     print  requests  that  were  stopped  by  lpshut.  See
     lpshut(1M).
```

Lab - Unable to print

```
# more /var/svc/log/application-print-server:default.log
[ Feb 11 16:10:24 Enabled. ]
[ Feb 11 16:10:24 Executing start method ("/usr/lib/lpsched") ]
Print services started.
[ Feb 11 16:10:24 Method "start" exited with status 0 ]
[ Feb 11 16:10:24 Stopping because all processes in service exited. ]
[ Feb 11 16:10:24 Executing stop method ("/usr/lib/lpshut") ]
Print services stopped.
[ Feb 11 16:10:25 Method "stop" exited with status 0 ]
[ Feb 11 16:10:36 Executing stop method ("/usr/lib/lpshut") ]
[ Feb 11 16:10:36 Method "stop" exited with status 0 ]
[ Feb 11 16:10:36 Executing start method ("/usr/lib/lpsched") ]
[ Feb 11 16:10:37 Method "start" exited with status 0 ]
[ Feb 11 16:10:37 Stopping because all processes in service exited. ]
[ Feb 11 16:10:37 Executing stop method ("/usr/lib/lpshut") ]
[ Feb 11 16:10:37 Method "stop" exited with status 0 ]
[ Feb 11 16:10:37 Disabled. ]
```

Lab - Unable to print

```
# file /usr/lib/lpsched
/usr/lib/lpsched:       executable /bin/ksh script
# more /usr/lib/lpsched
#!/bin/ksh
#
# ident "@(#)lpsched    1.2     04/11/01 SMI"
#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
        [ -f /usr/lib/lp/local/lpsched ] || exit 1
.
.
.
# Check to see if lpsched is already running
state=`svcprop -p restarter/state svc:/application/print/server:default`
```

Lab - Unable to print

If you dig deeper and spend some time looking at /usr/lib/lpsched you may notice the following lines in the script:

```
if [ "$OPTS" = "" ]  ; then
        /usr/sbin/svcadm enable -t svc:/application/print/server:default
        if [ $? = 0 ] ; then
                /bin/gettext "Print services started.\n"
                exit 0
        else
                exit 1
        fi
```

Lab - Unable to print

Something is clearly wrong with the start method as it is causing a loop.

```
# cd /var/svc/manifest/application/print
# more server.xml
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
...
        <exec_method
            type='method'
            name='start'
            exec='/lib/svc/method/print-svc start'
            timeout_seconds='60' />

        <exec_method
            type='method'
            name='stop'
            exec='/lib/svc/method/print-svc stop'
            timeout_seconds='60' />
```

Lab - Unable to print

1) Using setprop

```
# svccfg
svc:> select application/print/server
svc:.../server> setprop start/exec = "/lib/svc/method/print-svc start"
svc:.../server> setprop stop/exec = "/lib/svc/method/print-svc stop"
svc:.../server> quit
# svcadm refresh print/server:default
# svcadm -v enable print/server
svc:/application/print/server:default enabled.
# svcs print/server
STATE          STIME    FMRI
online         17:29:45 svc:/application/print/server:default
```

Lab - Unable to print

2) Re-importing the manifest

```
# svccfg
svc:> export svc:/application/print/server > /tmp/print.broken.xml
svc:> delete print/server
svc:> import /var/svc/manifest/application/print/server.xml
svc:> export svc:/application/print/server > /tmp/print.import.xml
```

Lab - Unable to print

If you look at the diff output the answer is clear:

```
# diff /tmp/print.import.xml /tmp/print.broken.xml
5c5
<     <create_default_instance enabled='false'/>
---
>     <create_default_instance enabled='true'/>
22c22
<     <exec_method name='start' type='method' exec='/lib/svc/method/print-svc start'
 timeout_seconds='60'>
---
>     <exec_method name='start' type='method' exec='/usr/lib/lpsched' timeout_seconds='60'>
25c25
<     <exec_method name='stop' type='method' exec='/lib/svc/method/print-svc stop'
 timeout_seconds='60'>
---
>     <exec_method name='stop' type='method' exec='/usr/lib/lpshut' timeout_seconds='60'>
29d28
<       <propval name='fd_limit' type='count' value='0'/>
32a32
>       <propval name='fd_limit' type='count' value='8192'/>
```

Lab - Unable to print

```
svc:.../server> setprop lpsched/fd_limit = 8192
svc:.../server> quit
# svcadm refresh svc:/application/print/server:default
# svcadm enable svc:/application/print/server:default
```